WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Friday, March 12, 2004

Hide?	Set Name	Query	Hit Count
	DB=EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
☐	L9	L8	2
☐	L8	L4	2
	DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ		
☐	L7	L6 and l3	18
☐	L6	L2 and l1	367
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
☐	L5	L4 and l2	11
☐	L4	L3 and wiring	267
☐	L3	object near interconnect$	1882
☐	L2	object and pointer and message	21898
	DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ		
☐	L1	717/100-109,116,120,165.ccls.	1505

END OF SEARCH HISTORY

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 18 of 18 returned.

☐ 1. Document ID: US 20030135850 A1

Using default format because multiple data bases are involved.

L7: Entry 1 of 18 File: PGPB Jul 17, 2003

PGPUB-DOCUMENT-NUMBER: 20030135850
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20030135850 A1

TITLE: System of reusable software parts and methods of use

PUBLICATION-DATE: July 17, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Miloushev, Vladimir I.	Laguna Niguel	CA	US	
Nickolov, Peter A.	Irvine	CA	US	

US-CL-CURRENT: 717/165

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | In

☐ 2. Document ID: US 20020120924 A1

L7: Entry 2 of 18 File: PGPB Aug 29, 2002

PGPUB-DOCUMENT-NUMBER: 20020120924
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020120924 A1

TITLE: System of reusable software parts for distributing event flows and methods of use

PUBLICATION-DATE: August 29, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Miloushev, Vladimir I.	Laguna Niguel	CA	US	
Nickolov, Peter A.	Irvine	CA	US	

US-CL-CURRENT: 717/165

ABSTRACT:

A system of reusable software parts for designing and constructing software components, applications and entire systems by assembly. Parts for generating events, shaping, distributing and controlling flows of events and other interactions are included. Also included are parts for handling synchronization and desynchronization of events and other interactions between parts, as well as parts for handling properties, parameterizing and serializing components, applications and systems. In addition, innovative adapter parts for interfacing parts that are not designed to work together are included. The system includes a dynamic container for software parts which supports integration of dynamically changing sets of parts into statically defined structures of parts. Other reusable parts for achieving such integration are also included.

☐ 3. Document ID: US 20020069400 A1

L7: Entry 3 of 18 File: PGPB Jun 6, 2002

PGPUB-DOCUMENT-NUMBER: 20020069400
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020069400 A1

TITLE: System for reusable software parts for supporting dynamic structures of parts and methods of use

PUBLICATION-DATE: June 6, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Miloushev, Vladimir I.	Laguna Niguel	CA	US	
Nickolov, Peter A.	Irvine	CA	US	

US-CL-CURRENT: 717/108

ABSTRACT:

A system of reusable software parts for designing and constructing software components, applications and entire systems by assembly. Parts for generating events, shaping, distributing and controlling flows of events and other interactions are included. Also included are parts for handling synchronization and desynchronization of events and other interactions between parts, as well as parts for handling properties, parameterizing and serializing components, applications and systems. In addition, innovative adapter parts for interfacing parts that are not designed to work together are included. The system includes a dynamic container for software parts which supports integration of dynamically changing sets of parts into statically defined structures of parts. Other reusable parts for achieving such integration are also included.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | In

☐ 4. Document ID: US 20020069399 A1

L7: Entry 4 of 18 File: PGPB Jun 6, 2002

PGPUB-DOCUMENT-NUMBER: 20020069399
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020069399 A1

TITLE: System of reusable software parts and methods of use

PUBLICATION-DATE: June 6, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Miloushey, Vladimir I.	Laguna Niguel	CA	US	
Nickolov, Peter A.	Irvine	CA	US	

US-CL-CURRENT: 717/108

ABSTRACT:

A system of reusable software parts for designing and constructing software components, applications and entire systems by assembly. Parts for generating events, shaping, distributing and controlling flows of events and other interactions are included. Also included are parts for handling synchronization and desynchronization of events and other interactions between parts, as well as parts for handling properties, parameterizing and serializing components, applications and systems. In addition, innovative adapter parts for interfacing parts that are not designed to work together are included. The system includes a dynamic container for software parts which supports integration of dynamically changing sets of parts into statically defined structures of parts. Other reusable parts for achieving such integration are also included.

☐ 5. Document ID: US 20020066076 A1

L7: Entry 5 of 18 File: PGPB May 30, 2002

PGPUB-DOCUMENT-NUMBER: 20020066076
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020066076 A1

TITLE: Object-oriented programming apparatus, object-oriented programming supporting apparatus, component builder apparatus, object-oriented program storage medium, program storage medium for use in object-oriented programming, component storage medium, and object-between-network display method

PUBLICATION-DATE: May 30, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Nagashima, Fumio	Kawasaki-shi		JP	
Suzuki, Kaori	Kawasaki-shi		JP	
Yumoto, Asako	Kawasaki-shi		JP	
Maruyama, Tsuguto	Kawasaki-shi		JP	

Sasaki, Shigeru	Kawasaki-shi	JP
Suda, Ryousuke	Kawasaki-shi	JP
Ueki, Miwa	Kawasaki-shi	JP

US-CL-CURRENT: 717/108; 717/116

ABSTRACT:

As to an object-oriented programming, reuse of softwares is enhanced and running speed is improved. There are made up a data element list in which pointers to data storage areas of object A are arranged and a pointer element list in which pointers to pointer storage areas of object B are arranged. A combination of the data element list and the pointer element list makes it possible to directly refer to data of the object A from the object B.

☐ 6. Document ID: US 20010052109 A1

L7: Entry 6 of 18 File: PGPB Dec 13, 2001

PGPUB-DOCUMENT-NUMBER: 20010052109
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20010052109 A1

TITLE: Object-oriented programming apparatus, object-oriented programming supporting apparatus, component builder apparatus, object-oriented program storage medium, program storage medium for use in object-oriented programming, component storage medium, and object-between-network display method

PUBLICATION-DATE: December 13, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Nagashima, Fumio	Kawasaki-shi		JP	
Suzuki, Kaori	Kawasaki-shi		JP	
Yumoto, Asako	Kawasaki-shi		JP	
Maruyama, Tsuguto	Kawasaki-shi		JP	
Sasaki, Shigeru	Kawasaki-shi		JP	
Suda, Ryousuke	Kawasaki-shi		JP	
Ueki, Miwa	Kawasaki-shi		JP	

US-CL-CURRENT: 717/100

ABSTRACT:

As to an object-oriented programming, reuse of softwares is enhanced and running speed is improved. There are made up a data element list in which pointers to data storage areas of object A are arranged and a pointer element list in which pointers to pointer storage areas of object B are arranged. A combination of the data element list and the pointer element list makes it possible to directly refer to data of the object A from

the object B.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Ir

---

☐ 7. Document ID: US 20010042239 A1

L7: Entry 7 of 18 File: PGPB Nov 15, 2001

PGPUB-DOCUMENT-NUMBER: 20010042239
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20010042239 A1

TITLE: Object-oriented programming apparatus, object-oriented programming supporting apparatus, component builder apparatus, object-oriented program storage medium, program storage medium for use in object-oriented programming, component storage medium, and object-between-network display method

PUBLICATION-DATE: November 15, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Nagashima, Fumio	Kawasaki-shi		JP	
Suzuki, Kaori	Kawasaki-shi		JP	
Yumoto, Asako	Kawasaki-shi		JP	
Maruyama, Tsuguto	Kawasaki-shi		JP	
Sasaki, Shigeru	Kawasaki-shi		JP	
Suda, Ryousuke	Kawasaki-shi		JP	
Ueki, Miwa	Kawasaki-shi		JP	

US-CL-CURRENT: 717/100

ABSTRACT:

As to an object-oriented programming, reuse of softwares is enhanced and running speed is improved. There are made up a data element list in which pointers to data storage areas of object A are arranged and a pointer element list in which pointers to pointer storage areas of object B are arranged. A combination of the data element list and the pointer element list makes it possible to directly refer to data of the object A from the object B.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Ir

---

☐ 8. Document ID: US 6604196 B1

L7: Entry 8 of 18 File: USPT Aug 5, 2003

US-PAT-NO: 6604196
DOCUMENT-IDENTIFIER: US 6604196 B1

TITLE: Apparatus and method for component role fulfillment based on environment context

DATE-ISSUED: August 5, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Monday; Paul Brian	Rochester	MN		
Rubin; Bradley Scott	Rochester	MN		

US-CL-CURRENT: 713/100; 717/108

ABSTRACT:

A method and apparatus for changing a component's mode based on its environmental context is disclosed. A component role fulfillment mechanism, as part of a component, changes the mode of a component based on its environmental context. The environmental context is defined by connections made by an assembler between components. Each component can support a variety of modes, but the component role fulfillment mechanism chooses one mode based on the environment and configures the component to fulfill only one mode.

56 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

☐ 9. Document ID: US 6557165 B1

L7: Entry 9 of 18 File: USPT Apr 29, 2003

US-PAT-NO: 6557165
DOCUMENT-IDENTIFIER: US 6557165 B1

TITLE: OBJECT-ORIENTED PROGRAMMING APPARATUS, OBJECT-ORIENTED PROGRAMMING SUPPORTING APPARATUS, COMPONENT BUILDER APPARATUS, OBJECT-ORIENTED PROGRAM STORAGE MEDIUM, PROGRAM STORAGE MEDIUM FOR USE IN OBJECT-ORIENTED PROGRAMMING, COMPONENT STORAGE MEDIUM, AND OBJECT-BETWEEN-NETWORK DISPLAY METHOD

DATE-ISSUED: April 29, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nagashima; Fumio	Kawasaki			JP
Suzuki; Kaori	Kawaski			JP
Yumoto; Asako	Kawasaki			JP
Maruyama; Tsuguto	Kawasaki			JP
Sasaki; Shigeru	Kawasaki			JP
Suda; Ryousuke	Kawasaki			JP
Ueki; Miwa	Kawasaki			JP

US-CL-CURRENT: 717/108; 707/203, 715/511, 717/116

ABSTRACT:

As to an object-oriented programming, reuse of softwares is enhanced and running speed is improved. There are made up a data element list in which pointers to data storage areas of object A are arranged and a pointer element list in which pointers to pointer storage areas of object B are arranged. A combination of the data element list and the pointer element list makes it possible to directly refer to data of the object A from the object B.

2 Claims, 132 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 84

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. Desc | Ir

---

☐ 10. Document ID: US 6539520 B1

L7: Entry 10 of 18 File: USPT Mar 25, 2003

US-PAT-NO: 6539520
DOCUMENT-IDENTIFIER: US 6539520 B1

TITLE: Systems and methods for generating hardware description code

DATE-ISSUED: March 25, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Tiong; Spencer Hao	Singapore			SG
Lim; Alvin Swee Hock	Singapore			SG

US-CL-CURRENT: 716/3; 707/101, 707/102, 707/104.1, 709/203, 715/513, 716/18, 717/106, 717/114, 717/140

ABSTRACT:

An Internet hardware description code generation system, methods, and scripts are provided. The Internet hardware description code generation system includes a hardware description code generation host adapted to generate one or more hardware description language files in response to one or more input parameters. A user uploads input parameters corresponding to a circuit to the hardware description code host. In response, the host generates one or more hardware description language (HDL) files that describe the circuit. Cgi scripts may be used to generate the HDL files.

36 Claims, 24 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 23

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. Desc | Ir

☐ 11. Document ID: US 6275977 B1

L7: Entry 11 of 18 File: USPT Aug 14, 2001

US-PAT-NO: 6275977
DOCUMENT-IDENTIFIER: US 6275977 B1

TITLE: Application cooperation method and apparatus

DATE-ISSUED: August 14, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nagai; Yasuhiko	Tokyo			JP
Nakayama; Yoshiyuki	Kawasaki			JP
Yumoto; Kazuma	Yokohama			JP
Hayashi; Masato	Kawasaki			JP
Yamane; Toshio	Yokohama			JP
Kinoshita; Shigeaki	Yokohama			JP

US-CL-CURRENT: 717/104; 717/108

ABSTRACT:

In an application cooperation apparatus for supporting configuration/reconfiguration of a business process support system, node class object information defining a logical AP object as an object class for each common process item and attribute and defining an inheritance relationship between object classes as a tree structure is provided to define interactively with a user a logical process model having logical AP objects as business components. Next, by referring to logical-physical object correspondence relationship information, an implementation process model for the logical process model is developed to generate execution control scripts in a definition language of this model.

16 Claims, 16 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 16

☐ 12. Document ID: US 6226792 B1

L7: Entry 12 of 18 File: USPT May 1, 2001

US-PAT-NO: 6226792
DOCUMENT-IDENTIFIER: US 6226792 B1

TITLE: Object management system supporting the use of application domain knowledge mapped to technology domain knowledge

DATE-ISSUED: May 1, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Goiffon; David A.	Shoreview	MN		
Hartmann; Gerald E.	Minneapolis	MN		
Johnson; David R.	Oakdale	MN		

US-CL-CURRENT: 717/120; 707/200

ABSTRACT:

An object management system is providing for managing, cataloging, and discovering various potentially reusable code and data components that exist within an Information Technology (IT) platform, and which each have well-defined interfaces with other components. For each of these re-usable code and data components, an associated software object called an "asset element" is created that describes the associated component. Relationships are created between various asset elements to represent the relationships existing between the software components. Other software objects called "locator elements" are created that each describes an application concept or sub-concept. This application concept or sub-concept is associated with a problem solved by the code and data components within the IT platform. Relationships are created between the various locator elements to correlate the concepts and sub-concepts to software constructs represented by asset elements. The object management system further supports various object discovery tools capable of identifying locator elements associated with a particular concept. These locator elements and the associated relationships may then be efficiently traced to identify related asset elements and the associated software and code constructs. This provides an efficient concept-based search mechanism for the code constructs. Other tools are provided for creating, modifying, and deleting the elements. A model may be used to define the various types of relationships and elements that may exist within the system, thereby simplifying the various tools needed to support element creation, modification, deletion, and traversal.

7 Claims, 24 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 20

☐ 13. Document ID: US 6223342 B1

L7: Entry 13 of 18 File: USPT Apr 24, 2001

US-PAT-NO: 6223342
DOCUMENT-IDENTIFIER: US 6223342 B1

TITLE: Object-oriented sequencing using hierarachical configuration streams

DATE-ISSUED: April 24, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
George; A. Chacko	Mountain View	CA	94040	

US-CL-CURRENT: 717/116

ABSTRACT:

A system and method for data processing objects having a number of attributes includes a sequence of object lists, each having data for identifying attribute values for at least one object. In a vector implementation, the object lists may be associated with a number of different objects. Each object list has a one-to-one correspondence between associated objects and local configuration steams that have configuration data indicative of the data structure for attributes of the object. In addition to the object lists, a global configuration stream is linked to form an object sequence. The global configuration stream includes configuration data indicative of the arrangement of the object lists. In a list implementation, each object list has linked smart pointers to locations in a memory pool that is used to store the attribute values. In the preferred embodiment, the object lists and/or the attribute smart pointers are indexed to increase traversal speed for enhanced performance.

13 Claims, 10 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 8

☐ 14. Document ID: US 6003037 A

L7: Entry 14 of 18 File: USPT Dec 14, 1999

US-PAT-NO: 6003037
DOCUMENT-IDENTIFIER: US 6003037 A

TITLE: Smart objects for development of object oriented software

DATE-ISSUED: December 14, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Kassabgi; George	Peabody	MA		
Sadd; John R.	Nashua	NH		
Wood; William T.	Boxborough	MA		

US-CL-CURRENT: 707/103R; 717/100, 717/116, 719/315

ABSTRACT:

An improved object-oriented programming environment for facilitating creation of database management applications is disclosed. The programming environment provides a method and apparatus for establishing named connections between encapsulated, individually designed software components referred to as "smart objects," which communicate and act in a coordinated fashion as part of a finished software application. Connections between smart objects are referred to as "smart links." A basic set of smart links for coordinating a core group of smart objects is disclosed, and a substantial variety of database management applications can be created using the disclosed smart objects and smart containers. The core group of smart objects each include four common capabilities: the capability to initialize and destroy themselves, the capability to get and set attributes, the capability to add and remove smart links, and the capability to communicate using a standard communication interface. Individual smart objects have

other capabilities in addition to the four common capabilities.

20 Claims, 15 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

☐ 15. Document ID: US 5680619 A

L7: Entry 15 of 18 File: USPT Oct 21, 1997

US-PAT-NO: 5680619
DOCUMENT-IDENTIFIER: US 5680619 A

TITLE: Hierarchical encapsulation of instantiated objects in a multimedia authoring system

DATE-ISSUED: October 21, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gudmundson; Norman K.	San Mateo	CA		
Forsythe; R. Hamish	Palo Alto	CA		
Lee; Wayne A.	San Mateo	CA		

US-CL-CURRENT: 717/108; 715/500.1, 715/515, 717/109, 717/111

ABSTRACT:

An application development system, optimized for authoring multimedia titles, enables its users to create selectively reusable object containers merely by defining links among instantiated objects. Employing a technique known as Hierarchical Encapsulation, the system automatically isolates the external dependencies of the object containers created by its users, thereby facilitating reusability of object containers and the objects they contain in other container environments. Authors create two basic types of objects: Elements, which are the key actors within an application, and Modifiers, which modify an Element's characteristics. The object containers (Elements and Behaviors--i.e., Modifier containers) created by authors spawn hierarchies of objects, including the Structural Hierarchy of Elements within Elements, and the Behavioral Hierarchy, within an Element, of Behaviors (and other Modifiers) within Behaviors. The system utilizes an Element's dual hierarchies to make that Element an environmental frame of reference to the objects it contains. Through techniques known as Hierarchical Message Broadcasting, Hierarchical Variable Scoping and Hierarchical Relative Positioning, objects automatically receive messages sent to their object container and access data known to their object container. An Element's position is even determined relative to the position of its parent Element container. The system is highly extensible through a Component API in which Modifiers and Services that support them can be added and integrated seamlessly into the system. The system's architecture is substantially platform-independent, automatically allowing most author's titles to run on multiple platforms. In addition, the entire authoring environment can be ported relatively easily to a variety of platforms due to the isolation a platform-dependent layer within the system.

48 Claims, 72 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 46

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. Desc | Ir

---

☐ 16. Document ID: US 5455952 A

L7: Entry 16 of 18 File: USPT Oct 3, 1995

US-PAT-NO: 5455952
DOCUMENT-IDENTIFIER: US 5455952 A

TITLE: Method of computing based on networks of dependent objects

DATE-ISSUED: October 3, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gjovaag; Inghard J.	Portland	OR		

US-CL-CURRENT: 717/108; 706/11

ABSTRACT:

An object dependency network-based data visualization system allows a user to graphically edit a network of displayed objects and their interconnections to specify an underlying data computing and data visualization process. Each displayed object is selected by the user from among a menu of objects representing data structures and functions such as data fields to be analyzed, mathematical operations, data input functions, and display manipulation functions. The user graphically draws lines interconnecting ports on the objects that have underlying data structure addresses that establish and control a data and operational flow through the underlying computer-driven process. The underlying process is continually running but only performs a particular computation in response to a data change or state change associated with any of the objects in the network. Thereby, an improved data visualization system is provided that minimizes memory usage and computation time while maintaining accuracy of the computed results, which is particularly beneficial when analyzing data derived from very large data structures such as scalar, vector, and tensor data fields.

13 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. Desc | Ir

---

☐ 17. Document ID: US 5396626 A

L7: Entry 17 of 18 File: USPT Mar 7, 1995

US-PAT-NO: 5396626

DOCUMENT-IDENTIFIER: US 5396626 A

TITLE: Object-oriented locator system

DATE-ISSUED: March 7, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nguyen; Frank T.	Campbell	CA		

US-CL-CURRENT: 717/171; 713/2, 717/165

ABSTRACT:

A method and system for adding components (documents, tools, fonts, libraries, etc.) to a computer system without running an installation program. A location framework is employed to locate components whose properties match those specified in a search criteria. The framework receives notification from the system when components whose properties match the search criteria are added to or removed from the system.

24 Claims, 11 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

☐ 18. Document ID: US 5379430 A

L7: Entry 18 of 18 File: USPT Jan 3, 1995

US-PAT-NO: 5379430
DOCUMENT-IDENTIFIER: US 5379430 A

TITLE: Object-oriented system locator system

DATE-ISSUED: January 3, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nguyen; Frank T.	Campbell	CA		

US-CL-CURRENT: 707/3; 713/2, 717/111, 717/113, 717/120

ABSTRACT:

A method and system for adding system components (documents, tools, fonts, libraries, etc.) to a computer system without running an installation program. A location framework is employed to locate system components whose properties match those specified in a search criteria. The framework receives notification from the system when system components whose properties match the search criteria are added to or removed from the system.

23 Claims, 11 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. Desc

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L6 and L3	18

Display Format: - Change Format

Previous Page Next Page Go to Doc#

Your search matched **1** of **1011253** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

object<and>message<and>wiring Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Use case maps for attributing behaviour to system architecture**
Buhr, R.J.A.;
Parallel and Distributed Real-Time Systems, 1996. Proceedings of the 4th International Workshop on , 15-16 April 1996
Pages:3 - 10

[Abstract] [PDF Full-Text (736 KB)] **IEEE CNF**

Your search matched **3** of **1011253** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

object<and>message<and>pointer Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Concert/C: supporting distributed programming with language extensions and a portable multiprotocol runtime**
Auerbach, J.S.; Gopal, A.S.; Russell, J.R.; Kennedy, M.T.;
Distributed Computing Systems, 1994., Proceedings of the 14th International Conference on , 21-24 June 1994
Pages:152 - 159

[Abstract] [PDF Full-Text (704 KB)] **IEEE CNF**

---

2 **Computer-assisted discrimination among malignant lymphomas and leukemia using immunophenotyping, intelligent image repositories, and telemicroscopy**
Foran, D.J.; Comaniciu, D.; Meer, P.; Goodell, L.A.;
Information Technology in Biomedicine, IEEE Transactions on , Volume: 4 , Issue: 4 , Dec. 2000
Pages:265 - 273

[Abstract] [PDF Full-Text (184 KB)] **IEEE JNL**

---

3 **Implementing references as chains of links**
Maisonneuve, J.; Shapiro, M.; Collet, P.;
Object Orientation in Operating Systems, 1992., Proceedings of the Second International Workshop on , 24-25 Sept. 1992
Pages:236 - 243

[Abstract] [PDF Full-Text (596 KB)] **IEEE CNF**

CiteSeer Find: object and message and pointer and Documents Citations

Searching for **object and message and pointer and wiring**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Amazon B&N Google (RI) Google (Web) CSB DBLP
No documents match Boolean query. Trying non-Boolean relevance query.
1000 documents found. **Only retrieving 125 documents (System busy - maximum reduced).** Retrieving documents... **Order: relevance to query.**

A Unifying Type-Theoretic Framework for Objects - Hofmann, Pierce (1993) (Correct) (24 citations)
Press 1 A Unifying TypeTheoretic Framework for **Objects** Martin Hofmann Benjamin Pierce Department of
www.cs.indiana.edu/pub/pierce/abstroop.ps.gz

Models for Asynchronous Message Handling - Langendoen, Bhoedjang, Bal (1997) (Correct) (4 citations)
mes sages. This occurs frequently in parallel **object**based and **object**oriented programming systems. In
Models for Asynchronous **Message** Handling Koen Langendoen Raoul Bhoedjang Henri
ftp.cs.vu.nl/pub/amoeba/orca_papers/ieee-concurrency97.ps.gz

An Object Calculus with Algebraic Rewriting - Compagnoni, Fernández (Correct)
An **Object** Calculus with Algebraic Rewriting Adriana
www.ens.fr/~maribel/papers/PLILP97.ps.gz

Statistical Learning, Localization, and Identification of.. - Hornegger, Niemann (1995) (Correct) (1 citation)
Learning, Localization, and Identification of **Objects** Joachim Hornegger and Heinrich Niemann The
www5.informatik.uni-erlangen.de/TeX/Literatur/ps-dir/1995/Hornegger95:SLL.ps.gz

Frames, Objects and Relations: Three Semantic.. - Norrie, Reimer.. (1994) (Correct)
Frames, **Objects** and Relations: Three Semantic Levels for
www.globis.ethz.ch/publications/docs/1994d-nrlrs-krdb.ps.gz

An Analytical Evaluation of Static Coupling Measures for Domain.. - Poels (1998) (Correct)
Evaluation of Static Coupling Measures for Domain **Object** Classes Geert Poels Research Assistant of the
the classes (i.e.static coupling) and the **message** passing between instances of the classes (i.e.
www.econ.kuleuven.ac.be/tew/academic/infosys/Members/Snoeck/ECOOP98-OOPM.ps

Geometry and Algebra of Multiple Projective Transformations - Heyden (1995) (Correct) (9 citations)
several dioeerent cases of reconstruction of 3D **objects** from a number of 2D images, obtained by
www.maths.lth.se/matematiklth/personal/andersp/publ/dissabs.ps

Using Classes As Specifications For Automatic Construction Of.. - Tyugu (1994) (Correct) (2 citations)
Kista, Sweden tyugu@it.kth.se It is shown how the **object**oriented programming paradigm has been combined
of methods for the class. Special computemessages are intro duced as requests for program
it.kth.se/labs/se/Reports/classes-as-spec.ps.Z

Correction of a Memory Management Method for Lock-Free Data.. - Michael, Scott (1995) (Correct) (5 citations)
deleted nodes not to be reused until no active **pointers** point to them. Also, most lockfree algorithms
hypatia.dcs.qmw.ac.uk/data/edu/cs.rochester.edu/systems/95.tr599.Memory_management_for_lock-free_data_structures.ps.gz

Foreign Event Handlers to Maintain Information Consistency and.. - Queloz (1999) (Correct)
dynamicity of the world itself: people are moving, **objects** are created, exchanged, destroyed, new books are
allows communication with less conventions than **message** passing [5, 3]Processes interconnected by
cuiwww.unige.ch/~queloz/papers/mac3.1999.ps.gz

Performance Evaluation and Modeling of MPI Communications .. - Folino, Spezzano, Talia (Correct)
is proposed. It is based on the size of **messages** exchanged and the number of processors
isi-cnr.deis.unical.it:1080/~talia/hpcn98.ps

Mechanisms and Interfaces for Software-Extended Coherent Shared.. - Chaiken (1994) (Correct) (3 citations)

performance if their subtasks share large data **objects** and communicate data in bursts. Recent
a memory system. 56 52 Sample protocol **message** handler.
shared memory. The architecture with a fivepointer LimitLESS di rectory achieves between 71% and
ftp.cag.lcs.mit.edu/pub/papers/chaiken-dissert-1-10.ps.Z

Using PVM 3.0 to Run Grand Challenge Applications on.. - Dongarra, Geist.. (1992) (Correct)
configuration, dynamic process groups, multiple **message** buffers, process signalling, and user definable
be implemented by memory copies or by passing of **pointers** and careful use of locks. PVM 3.0 contains a
ftp.netlib.org/ncwn/siam93-pvmgc.ps

A Hypertext System for Integrating Heterogeneous, Autonomous.. - Noll, Scacchi (1994) (Correct) (2 citations)
into a graph structure of linked container **objects**. This paper ex amines issues involved in
local **objects** into DHT nodes and links, and DHT **messages** into local operations (see fig ure 2)From the
cwis.usc.edu/dept/ATRIUM/Papers/Integrating_Software_Repositories.ps

A Nonprehensile Method for Reliable Parts Orienting - Zumel (Correct) (3 citations)
then, can be defined as the manipulation of **objects** without grasping them. Manipulation without
pecan.srv.cs.cmu.edu/afs/cs.cmu.edu/misc/mosaic/common/omega/Web/People/mlab/papers/nbz-summary.ps

Probabilistic Object Recognition using Multidimensional Receptive .. - Schiele (1996) (Correct) (22 citations)
Probabilistic **Object** Recognition using Multidimensional Receptive
www-white.media.mit.edu/people/bernt/Pubs/icpr96.ps.gz

Learning Planning Operators by Observation and Practice - Wang (1994) (Correct) (12 citations)
for the domain, which includes the types of **objects** and the predicates that describe states and
www.rpal.rockwell.com/~mei/aips94.ps

[Recommendation X.904: - Basic Reference (1994) (Correct)
in LOTOS. 6.1.1 Basic Modelling Concepts 6.1.1.1 **Object**: An instantiation of a LOTOS process definition
ftp.gte.com/pub/odp/1994/part4_p2.ps.gz

A Meta-theory for Structured Presentations in the COC - Shulman (1997) (Correct)
What is a suitable theory to discharge a context **object** C? 150 6.2 Additional relationships :
cse.ogi.edu/pub/tech-reports/1997/97-TH-001.ps.gz

Scheduling Access To Temporal Data In Real-Time Databases - Xiong, Sivasankaran.. (1997) (Correct) (3 citations)
possibly restarted) when any of the temporal data **objects** they read become invalid before the transaction
www-ccs.cs.umass.edu/~sim/rtdb-chapter96.ps

First 20 documents Next 20

Try your query at: Amazon Barnes & Noble Google (RI) Google (Web) CSB DBLP

CiteSeer Find: interconnect and message and pointe Documents Citations

Searching for **interconnect and message and pointer and wiring**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Amazon B&N Google (RI) Google (Web) CSB DBLP
No documents match Boolean query. Trying non-Boolean relevance query.
1000 documents found. **Only retrieving 500 documents (System busy - maximum reduced).** Retrieving documents... **Order: relevance to query.**

Models for Asynchronous Message Handling - Langendoen, Bhoedjang, Bal (1997) (Correct) (4 citations)
models on a parallel ma chine with a fast **interconnect**. Implementation overview The nodes in our
Models for Asynchronous **Message** Handling Koen Langendoen Raoul Bhoedjang Henri
ftp.cs.vu.nl/pub/amoeba/orca_papers/ieee-concurrency97.ps.gz

A Unifying Type-Theoretic Framework for Objects - Hofmann, Pierce (1993) (Correct) (24 citations)
programming, including objects, methods, **message** passing, and subtyping, by introducing an
Abadi, Eugenio Moggi, and Andre Scedrov supplied **pointers** to relevant literature. Terry Stroup and
www.cs.indiana.edu/pub/pierce/abstroop.ps.gz

Correction of a Memory Management Method for Lock-Free Data.. - Michael, Scott (1995) (Correct) (5 citations)
deleted nodes not to be reused until no active **pointers** point to them. Also, most lockfree algorithms
hypatia.dcs.qmw.ac.uk/data/edu/cs.rochester.edu/systems/95.tr599.Memory_management_for_lock-free_data_structures.ps.gz

Performance Evaluation and Modeling of MPI Communications .. - Folino, Spezzano, Talia (Correct)
is proposed. It is based on the size of **messages** exchanged and the number of processors
isi-cnr.deis.unical.it:1080/~talia/hpcn98.ps

Mechanisms and Interfaces for Software-Extended Coherent Shared.. - Chaiken (1994) (Correct) (3 citations)
an extremely high bandwidth and expensive **interconnect**ion network, and might work well for
a memory system. 56 52 Sample protocol **message** handler.
shared memory. The architecture with a fivepointer LimitLESS di rectory achieves between 71% and
ftp.cag.lcs.mit.edu/pub/papers/chaiken-dissert-1-10.ps.Z

Using PVM 3.0 to Run Grand Challenge Applications on.. - Dongarra, Geist.. (1992) (Correct)
configuration, dynamic process groups, multiple **message** buffers, process signalling, and user definable
be implemented by memory copies or by passing of **pointers** and careful use of locks. PVM 3.0 contains a
ftp.netlib.org/ncwn/siam93-pvmgc.ps

Performance Comparison Of Video Transport Over ATM.. - Hossain, Kang, Horst (Correct)
Of Video Transport Over ATM &ServerNet **Interconnects** Ashfaq Hossain, SungMo Kang, Bob Horst Depts
rate adjustment feedbacks and repairrequest **messages** to the server after a video packet loss has been
berserk.vlsi.uiuc.edu/people/ashfaq/ieee.mm97.ps

Foreign Event Handlers to Maintain Information Consistency and.. - Queloz (1999) (Correct)
than **message** passing [5, 3]Processes **interconnect**ed by Mobile Code still have to agree on high
allows communication with less conventions than **message** passing [5, 3]Processes **interconnect**ed by
cuiwww.unige.ch/~queloz/papers/mac3.1999.ps.gz

Intelligent Computing About Complex Dynamical Systems - Zhao (1994) (Correct)
www.cis.ohio-state.edu/insight/papers/mcs.ps

Learning Planning Operators by Observation and Practice - Wang (1994) (Correct) (12 citations)
www.rpal.rockwell.com/~mei/aips94.ps

Using Classes As Specifications For Automatic Construction Of.. - Tyugu (1994) (Correct) (2 citations)
of methods for the class. Special computemessages are intro duced as requests for program
it.kth.se/labs/se/Reports/classes-as-spec.ps.Z

Formalising Abilities and Opportunities of Agents - van Linder, van der Hoek, Meyer (1998) (Correct) (2 citations)
different, abilities and opportunities are **interconnect**ed in that abilities can be exercised only when
ftp.cs.uu.nl/pub/RUU/CS/techreps/CS-1998/1998-08.ps.gz

Talking Vs Taking: Speech Access To Remote Computers - Yankelovich (1994) (Correct) (2 citations)
there that you forgot to print out the mail **message** with all the location information? For times
www.sunlabs.com/research/speech/publications/chi94/CHI94Short.ps

An Overview of Document Mining Technology - Dixon (1997) (Correct)
supplying the system with a MRD for that domain. **Message** Understanding Conference Over the last several
www.geocities.com/~mjdixon/mark/writings/dixm97_dm.ps

Development, Learning and Evolution in Animats - Kodjabachian, Meyer (1994) (Correct) (2 citations)
of forming a locomotion controller by fully **interconnect**ing six individual legcontrollers, took
set to 1. Squares represent input/output **pointer** cells. Continuous connections have a weight of
www.biologie.ens.fr/fr/animatlab/perso/kodjaba/jkjamperac.ps.gz

An Object Calculus with Algebraic Rewriting - Compagnoni, Fernández (Correct)
find =s)x:if s:text:cursor =x then s:**message** :found'else if s:text:cursor 6= then
www.ens.fr/~maribel/papers/PLILP97.ps.gz

Statistical Learning, Localization, and Identification of.. - Hornegger, Niemann (1995) (Correct) (1 citation)
www5.informatik.uni-erlangen.de/TeX/Literatur/ps-dir/1995/Hornegger95:SLL.ps.gz

Fourth And Fifth Order Efficiency: Fisher Information - Kano (Correct)
koko15.hus.osaka-u.ac.jp/members/kano/research/./dvi/fisher.ps

Nozomi - A Fast, Memory-Efficient Stack Decoder For Lvcsr - Schuster (1996) (Correct)
equivalence it can be linked into the lattice. A **pointer** on the best arc back has to be updated to not
www.aist-nara.ac.jp/IS/Shikano-lab/staff/1996/mike-s/papers/icslp98.ps.gz

FREE JAZZ: A User-Level Real-Time Threads Package Designed for.. - Kramp (1998) (Correct)
in the case of interthread communica tion, all **message** queues must be made explicit in such a way that
www.uni-kl.de/AG-Nehmer/Projekte/Squirrel/postscript/tr-sfb501-9-98.ps.gz

First 20 documents Next 20

Try your query at: Amazon Barnes & Noble Google (RI) Google (Web) CSB DBLP